Software Maintenance from a Service Perspective[*]

FRANK NIESSINK[1]** and HANS VAN VLIET[1]
[1]*Faculty of Sciences, division of Mathematics and Computer Science, Vrije Universiteit, De Boelelaan 1081a, 1081 HV Amsterdam, the Netherlands.*

SUMMARY
In this paper we investigate the differences between software maintenance and software development from a service point of view, and the consequences thereof for the maturity of software maintenance organizations. We argue that software maintenance can be seen as providing a service, whereas software development is concerned with the development of products. Differences between products and services affect the way in which customers assess their respective quality. Consequently, customers will judge the quality of software maintenance differently from that of software development. This in turn implies a need for different processes than those used by the average software development organization. We discuss two overall approaches to achieve a high-quality IT service organization which include these service-specific processes: ITIL and the IT Service Capability Maturity Model. ITIL is a set of best practices widely used within the IT service industry. The IT Service CMM is a maturity growth model akin to the Software CMM. © 2000 by John Wiley & Sons, Ltd.

J. Soft. Maint., **12**(2), 103-120, (2000)

No. of Figures: 3. No. of Tables: 0. No. of References: 22.

KEY WORDS: service versus product; IT services; quality assessment; process improvement; capability maturity model.

Contract Grant Sponsor: Dutch Ministry of Economic Affairs, project ‘KWINTES’, nr. ITU-96024.

**Address for Correspondence: Frank Niessink, Faculty of Sciences, division of Mathematics and Computer Science, Vrije Universiteit, De Boelelaan 1081a, 1081 HV Amsterdam, the Netherlands. Email: F.Niessink@cs.vu.nl.

1 INTRODUCTION

In this paper we investigate the differences between software maintenance and software development from a service point of view. We argue that there are differences between products and services in general. These differences affect the way in which customers assess the quality of products and services. In particular, service quality is assessed on two dimensions: the technical quality—what the result of the service is—and the functional quality—how the service is delivered.

We argue that software maintenance can be seen as providing a service, whereas software development is concerned with the development of products. Consequently, customers will judge the quality of software maintenance differently from that of software development. This means that to deliver high quality results in software maintenance, both the functional quality and the technical quality dimension

[*] This is a preprint of an article published in the *Journal of Software Maintenance: Research and Practice*, volume 12, number 2, March/April 2000, pp. 103-120.

are important. In order to provide high-quality software *maintenance*, different and additional processes are needed than provided by a high-quality software *development* organization.

We have been involved in a multi-partner research effort in which we aimed to develop a method to specify and control IT services. In the course of this research, a number of case studies were done to introduce and test parts of this framework in different organizations. These case studies had mixed results. We observed that some service providers were more mature as regards their service capabilities than others. Based on these experiences, we developed an IT Service Capability Maturity Model which provides software maintenance organizations, and other IT service providers, with a maturity model that focuses on services rather than on products. This IT Service CMM is discussed in Niessink and van Vliet (1998). The present paper concentrates on the service aspects of software maintenance and motivates our choice for specific elements of the maturity model.

This paper is structured as follows: in section 2, we discuss a number of differences between services and products in general, and between software maintenance and software development in particular. In section 3, we show how these differences affect organizations that maintain software. In particular, we identify a number of possible gaps between the service expected by the customers and the service they actually get. In section 4 we report on some case studies we did in the course of our research, and relate the results thereof to the gaps identified in section 3. In section 5 we describe processes to bridge these gaps. Section 6 discusses related work in assessing software maintenance quality. Section 7 discusses two overall approaches to achieving quality in IT services that can also be applied to software maintenance. Both these models incorporate the processes identified in section 3. Finally, section 8 presents our conclusions.

2 SERVICE VERSUS PRODUCT

In the service marketing literature, a wide range of definitions exists of what a service entails. Usually, a service is defined as an essentially intangible set of benefits or activities that are sold by one party to another (Grönroos, 1990, p. 27). The main differences between products and services are (Zeithaml, 1996):

- **Intangibility.** This is considered to be the most basic difference between products and services. Services—being benefits or activities—cannot be seen, felt, tasted, or touched, like products can. Consequently,
  - services cannot be inventoried,
  - services cannot be patented,
  - services cannot be readily displayed or communicated, and
  - pricing is more difficult.
- **Heterogeneity.** Because services are created by activities, and activities are performed by humans, services tend to be more heterogeneous than products. Consequently,
  - service delivery and customer satisfaction depend on employee actions,
  - service quality depends on factors which are difficult to control, such as the ability of the customer to articulate his or her needs, the ability and willingness of personnel to satisfy those needs, the presence or absence of other customers, and the level of demand for the service, and
  - these complicating factors make it hard to know whether the service was delivered according to plan or specifications.

- **Simultaneous Production and Consumption.** Services are produced and consumed simultaneously, whereas for products production and consumption can be separated. For example, a car can be produced first, sold a few months later, and then be consumed over a period of several years. For services on the other hand, the production and consumption has to take place in parallel. The production of the service creates the 'set of benefits', whose consumption cannot be postponed. For example, a restaurant service—preparing a meal, serving the customer—has largely to be produced while the customer is receiving the service. Consequently,
  - customers participate in and affect the transaction,
  - customers may affect each other, for example, noisy people in a restaurant may make dinner less enjoyable for other guests,
  - employees affect the service outcome, and
  - centralization and mass production are difficult.
- **Perishability.** Services cannot be saved or stored. Consequently,
  - it is difficult to synchronize supply and demand with services, and
  - services cannot be returned or resold.

The difference between products and services is not clear-cut. Often, services are augmented with physical products to make them more tangible, for example, luggage tags provided with a travel insurance. In the same way, products are augmented with add-on services, for example a guarantee, to improve the quality perception of the buyer. In the service marketing literature, e.g. Berry and Parasuraman (1991), a product-service continuum is used to show that there is no clear boundary between products and services. This product-service continuum is a spectrum with pure products on one end and pure services on the other end, and product-service mixtures in between. Figure 1 shows some example products and services positioned on the product-service continuum.

As Figure 1 shows, products and services can be intertwined. In the case of fast-food, both the product—the food itself—and the service—fast delivery—are essential to the customer. This means that the quality of such a product-service mix will be judged on both product and service aspects: is the food quickly served, and does it taste well.

If we turn to the software engineering domain, we see that a major difference between software development and software maintenance is the fact that software development results in a *product*, whereas software maintenance results in a *service* being delivered to the customer. Software maintenance is defined as (IEEE, 1990): "The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment". Usually, four types of maintenance are distinguished (Lientz and Swanson, 1980):

1. **Corrective maintenance** deals with the repair of faults found.
2. **Adaptive maintenance** deals with adapting the software to changes in the environment, such as new hardware or the next release of an operating system. Adaptive maintenance does not lead to changes in the system's functionality.
3. **Perfective maintenance** mainly deals with accommodating new or changed user requirements. It concerns functional enhancements to the system.
4. **Preventive maintenance** concerns activities aimed at increasing the system's maintainability, such as updating documentation or adding comments.

All these types of maintenance are concerned with activities aimed at keeping the system usable and valuable for the organization. So, software maintenance has more service-like aspects than software development, because the value of software maintenance is in the activities that result in benefits for the customers, such as corrected faults and new features. Contrast this with software development, where the development activities do not provide benefits for the customer, but instead it is the resulting software system that provides the benefits.

As said above, the difference between products and services is not clear-cut. Consequently, this goes for software development and software maintenance as well. Figure 2 shows the product-service continuum, as displayed in Figure 1, but with examples from the software engineering domain.

3 SERVICES AND QUALITY

Though we argued in the previous section that we can view software maintenance as a service and software development as product development, we did not yet discuss why this would be beneficiary. In order to do so, we again turn to the literature in the area of service management and marketing.

Grönroos (1990, p. 37) states that there are two dimensions that determine the experienced quality of services:

- The technical quality of the outcome. This dimension is formed by the result of the service, *what* the customer is left with when the service has been delivered.
- The functional quality of the process. This dimension is determined by the way in which the customer receives the service, in other words *how* the service is delivered.

Service marketeers often use the gap model to illustrate how differences between perceived service delivery and expected service can come about, see Figure 3. The net difference between the perceived quality of the services and the expected quality (gap 5) is caused by four other gaps (Zeithaml and Bitner, 1996):

Gap 1. The expected service as perceived by the service provider differs from the service as expected by the customer.
Due to inadequate market research, lack of communication between contact employees and management, and insufficient relationship focus, the service provider has a perception of what the customer expects which differs from the real expected service.
For example, the service organization aims to satisfy certain availability constraints (e.g. 99% availability), while the actual customer concern is with maximum downtime (e.g. no longer than one hour per failure).

Gap 2. The service specification as used by the service provider differs from the expected service as perceived by the service provider.
Caused by a lack of customer-driven standards, absence of process management, lack of a formal process for setting service quality goals, poor service design and inadequate service leadership, the service designs and standards will not match the service requirements as perceived by the provider.
For example, the customer expects a quick restart of the system, while the standard procedure of the maintenance organization is focused on analyzing the reason for the crash.

Gap 3. The actual service delivery differs from the specified services.
Service delivery does not follow the service designs and standards because of deficiencies in human resource policies, failures to match demand and supply, and customers not fulfilling their role.

For example, customers bypass the helpdesk by phoning the maintainer of their system directly, and thus hinder a proper incident management process.

Gap 4. Communication about the service does not match the actual service delivery. Communication by the service provider about its delivered services does not match the actual service delivery because of ineffective management of customer expectations, overpromising, and inadequate horizontal communications (i.e. insufficient communication between sales and operations and between advertising and operations and differences in policies and procedures across the organization). For example, a customer is not informed about the repair of a bug he or she reported.

The fifth gap is caused by the four preceding gaps. Hence, perceived service quality can be increased by closing the first four gaps, thus bringing the perceived service in line with the expected service.

To summarize so far, we see that the quality of services is determined by two quality dimensions: the technical quality—what is the result—and the functional quality—how is the result reached. We also showed how the gap between the perceived service delivery and expected service delivery is caused by several other gaps in the service provider's organization.

The question is, how does this all translate to the area of software engineering? Our argument is that since software maintenance organizations are essentially service providers, they need to consider the issues mentioned in this section. They need to manage their product—software maintenance—as a service to be able to deliver high quality software maintenance.

Looking at the gap model presented in Figure 3, we notice a number of processes emerge which pertain to the quality of the delivered services. To close the gaps a service provider needs to:

1. Translate customer service expectations into clear service agreements (Gap 1).
2. Use the service agreements as a basis for planning and implementing the service delivery (Gap 2).
3. Ensure that service delivery is done according to planning and procedures (Gap 3).
4. Manage communication about the services delivered (Gap 4).

4 CASE STUDIES

In this section we present some of the case studies that were done in the course of our research project and have particular reference to service quality. In this project, different research issues have been identified, including the specification of service level agreements (SLA) (Trienekens, van der Zwan, and Bouman, 1999), evaluation of service quality, the use of service catalogs and problem management. These issues have been investigated in several case studies. The three case studies presented below illustrate the occurrence of some of the gaps identified above.

4.1 Case A: Developing a Service Level Agreement

This case study was part of an education improvement program undertaken by a Dutch university. Part of the program is the supply of notebooks to all students. The first notebooks were offered to the 1997/1998 generation, which means that in the year 2002 every student at the university will have a notebook. It is expected that at that time more than 5000 notebooks will be in use. The notebooks are used in lectures, during tutorials, in study groups, and at home.

In order to make the improvement program successful, the university has to provide IT services to the students such as end-user support and repair maintenance. To delivery these services, a new service centre was created. This Notebook Service Centre solves hardware and software problems, installs new software and provides user support for the students.

During the case study a service level agreement between the Notebook Service Centre and the university was developed. The SLA specification method developed in the course of our project was used to derive the needed service levels, taking the students—the end-users—as the starting point. The specification process consisted of two steps: during the first step, the needed services and service levels were identified. During the second step, a service level agreement was written. The main services identified were the availability of the notebooks and the user support. For each of these services, measurable service levels were negotiated and put down in the service level agreement.

Despite the lack of experience with service level agreements of the participating parties in this case study, a SLA was successfully established, using the SLA specification method to structure the process.

4.2 Case B: Developing a Service Catalog

This case study was done in the central IT department of a large Dutch governmental organization. The IT department develops, operates, and maintains hardware and software for the decentralized governmental organization. The IT department supports and maintains 75 information systems and 45,000 workstations. The goal of the case study was to investigate the possibility for using a service catalog to improve communication between the IT department and its customers. The purpose of the service catalog would be to facilitate the negotiation of service levels by providing a set of services combined with standard service levels that the IT department is able to provide, together with standard prices.

When the case study started, the IT department had already developed a document that was supposed to be a service catalog. This document contained a list of information systems exploited by the IT department, and lists of services provided by the IT department, such as office automation services, batch services, and system development services. However, closer investigation showed that this document did not contain the information necessary to negotiate service levels: it hardly contained any quantitative data and no indications of costs of services.

Further research showed that the organization did not only omit this information from the service catalog, but also that it did not have the necessary data available. The IT department does have service level agreements with its customers, but these SLAs need to be improved. Often, the agreements do not contain quantitative service levels, or the agreements are formulated in terms of effort obligations (‘We will investigate the incident within one hour’) instead of in terms of result obligations (‘We will solve the incident within one hour or provide a work-around’). Hence, little detailed quantitative data is available on the real service levels that the IT department delivers.

The described situation made it impossible to implement a full scale service catalog during the time-span of this case study.

4.3 Case C: Incident and Problem Management

This organization is the IT department of a large organization, responsible for carrying out part of the Dutch social security system. As of the beginning of 1996, the organization has been split into a non-profit public body and a private for-profit organization—part of which is the IT department.

The IT department provides a large number of IT services to its customers, which are mainly departments from the sibling organization. To manage the communication with customers regarding those services, the department has implemented helpdesk management and problem management processes. The implementation of these processes has been based on the Information Technology Infrastructure Library (ITIL). The ITIL process Helpdesk Management is used to guarantee the continuity of services, while the ITIL process Problem Management is used to improve the level of service in the future. So, Helpdesk Management deals with *incidents*, whereas Problem Management is concerned with solving the *problems* that cause these incidents.

The goal of this case study was to assess the quality of the Problem Management process. As a first step, an assessment was made of the validity of the helpdesk database in which the incidents are recorded. 200 incidents were randomly selected, and the incidents were classified by a helpdesk employee, based on the free text description of the incidents. Often, the classification was impossible because the free text description of the incident was too vague. In the cases were a classification was possible, 30% of the new classifications differed from the existing classification. Of the 30,000 incidents in the helpdesk database, 56% was not classified at all.

It became apparent that the organization was not able to execute the Problem Management process properly, because the Helpdesk Management process did not result in the necessary data needed to adequately analyze and solve problems. This is illustrated by the fact that the helpdesk database contained only 110 identified problems, compared to 30,000 incidents. It was found necessary to first implement a clear and consistent registration of the incidents that occur during service delivery, before attempting to improve the problem management process.

4.4 Lessons learned

Although the three case studies discussed cover different issues and different organizations, we feel that several important lessons can be learned from these, and other case studies that we did. The most important lesson is that IT service improvement can only be successful if the organizational preconditions have been fulfilled.

The case studies were aimed at testing service improvement techniques and methods developed in the course of our project. On hindsight, the success, or lack of success, of the application of these techniques and methods can be easily interpreted in terms of the gap model. In particular:

- Gap 1 was successfully bridged in case A. The SLA specification method is specifically aimed at guiding the dialogue between customer and service provider as to the contents of the service. The resulting service level agreement provided a solid basis for planning and implementing the service.
- In case B, the organization wanted to create a service catalog to guide communication about its services, thus narrowing gap 4. However, the jump to a full-fledged service catalog was impossible without the necessary experience with quantitative, result-oriented service level agreements.
- Gap 4 was very visible in case C. Incident management was not done consistently, and this caused problems in the (internal as well as external) communication about the services delivered.

5 BRIDGING THE GAPS

In this section, we discuss four processes that may help software maintenance organizations bridge the gaps identified in section 1. Obviously, these processes were not derived straightforwardly from these gaps. What we report on here is the a posteriori justification of characteristics that distinguish a successful software maintenance organization from a less successful one, based on many discussions with maintenance managers, experiences with successful and less successful case studies, and a critical look at the Software Capability Maturity Model from a maintenance perspective.

5.1 Gap 1: Management of Commitments

It is important that maintenance commitments be planned and documented. This works best if the maintenance organization and customer work together towards the specification of relevant and realistic maintenance service commitments (often called Service Level Agreement—SLA), based on the needs of the customer. The actual maintenance services delivered, the specified service levels and the customer's service needs are reviewed with the customer on a regular basis. As a result of this evaluation, the service level agreement may have to be adjusted to stay in line with possibly changing maintenance needs.

There are two basic issues involved here: first, the maintenance service to be delivered is specified in a contract—the service level agreement—containing *measurable* service levels. Second, the service levels specified should address the business needs of the customer.

The service level agreement documents the maintenance services to be delivered. It covers the purpose, scope and goals of the services, their specification, and other agreements. The service level agreement functions as a means to close gap 1 by setting expectations for the maintenance service. It should at a minimum specify:

1. the maintenance services itself, i.e. a specification of the services to be delivered;
2. with what levels of service, i.e. how fast, how reliable, etc., specified in a measurable manner. Service levels need to be measurable because the organization has to report the realized service levels.
3. the conditions the customer should obey. Examples of such conditions could be that the customer should use a certain format for documenting change requests or, in case of a bug, provide the maintenance department with the input that caused the fault to manifest itself.
4. what happens if the maintenance organization does not reach the agreed upon service levels while the customer did not violate the customer conditions.
5. when and what will be reported to the customer regarding the actual delivered maintenance services.
6. when and how the service level agreement will be reviewed.
7. under which circumstances (calamities) service is not guaranteed.

The service commitments as documented in the service level agreement should be derived from the maintenance needs of the customer (as opposed to just the capabilities of the maintenance organization). These maintenance needs should be related to the business processes of the customer, its information technology, its business strategy, etc. This ensures that the maintenance organization thinks about what the customer needs and thus helps to close gap 1.

5.2 Gap 2: Maintenance Planning

The maintenance activities as specified in the service level agreement have to be planned. This includes the planning of the maintenance activities themselves, the transfer of the results thereof to the customer, the estimation of resources needed, the scheduling of maintenance activities, and the identification of possible risks.

In a normal, non-emergency situation, changes are often bundled into releases. There are various ways of deciding on the contents and timing of the next release. For example, releases may be scheduled at fixed time intervals, while there also is a fixed number of people available for doing maintenance. The next release will then contain all changes which could be handled within that time frame. One may also negotiate and fix the contents of the next release in advance, and allocate the number of people accordingly. This planning of releases is part of the maintenance planning process. See Stark and Oman (1997) for an observation of several strategies applied in practice.

In emergency situations, capacity needs to be available to react to the incident at hand. The incident needs to be solved or a work-around needs to be created for the customer.

Explicitly basing the planning of maintenance activities on the commitments as agreed upon with the customer helps to close gap 2.

5.3 Gap 3: Maintenance Activity Tracking

The service level agreement states which maintenance activities are to be carried out, and how fast, reliable, etc. this should be done. In order to be able to report on the performance of the maintenance organization in this respect, information about the actual maintenance activities is to be gathered. The purpose of the maintenance activity tracking process is to provide this information, monitor maintenance activities, and take corrective actions if necessary.

For example, when the customer reports a bug, information about the bug itself (originator, type, etc.) is recorded, as well as the reporting time, the time when corrective action was started and ended, and the time when the bug was reported fixed. If these data indicate that the average downtime of a system exceeds the level as specified in the service level agreement, the maintenance organization might assign more maintenance staff to this system, put maintenance staff on point-duty at the customer site, renegotiate the agreed upon service level, or take any other action to realign agreement and reality.

By keeping a strict eye upon the performance of the maintenance organization, and adjusting the maintenance planning and/or renegotiating the commitments with the customer when required, gap 3 is narrowed.

5.4 Gap 4: Event Management

Event management concerns the management of events that cause or might cause the maintenance activities carried out to deviate from the agreed upon levels of maintenance service. Events can be either:

- Requests for changes from users or other stakeholders. For example, requests for a new feature in the software;
- Incidents that cause or will cause service levels to be lower than agreed upon if no action is being taken. For example, a server that is down might cause the specified maximum down-time to be exceeded if it is not restarted quickly enough.

The main purpose of event management is to manage all events that occur during software maintenance. Event management encompasses a number of activities that should ensure that incidents and change requests are resolved in time and that affected groups, including the customer, are kept informed. These activities thus contribute to both the functional as well as the technical quality of software maintenance. A subset of the event management activities is:

- An event management library system is established as a repository for the event records.
  This event management library system (often in the form of a ‘helpdesk system’) should provide for the storage, update, and retrieval of event records, the sharing and transfer of event records between affected groups, and should help in the use of event management procedures.
  This supports the communication with the customer about the maintenance services delivered. It also supports the maintenance department itself, in its role of a historical data base of changes. The event management system thus helps to close gap 4.
- Events are identified, recorded, reviewed, and tracked according to a documented procedure.
  Each event is recorded in the library system, the impact of the event is assessed and documented, and ‘action items’ are formulated and initiated to resolve the event.
  This activity reinforces that the maintenance activities carried out are kept in accordance with the maintenance commitments and the maintenance planning, thus helping to close gap 3.
- Standard reports documenting the event management activities and the contents of the event repository are developed and made available to affected groups and individuals.
  This activity helps keeping the customer informed about the progress of activities to resolve incidents or process change requests. The communication with the customer not only pertains to individual change requests, but also their bundling into releases. There thus is a relation with the maintenance planning process. Keeping the customer informed will help manage customer expectations, again narrowing gap 4.

6 RELATED WORK

In his book *Practical Software Maintenance*, Thomas Pigoski laments that software maintenance organizations need to realize that they are in the customer service business (Pigoski, 1996, pp. 171-172). Apparently, this is not widely recognized yet. Within the software engineering domain, including software maintenance, the focus is on product aspects. The final phases of software development supposedly concern the delivery of an operations manual, installing the software, handling change requests and fixing bugs. In practice, the role of the IS department is much broader during the deployment stage, as illustrated by the ubiquitous help desk.

Published evaluations of software maintenance practices tend to concentrate on the narrow issue of efficiently handling change requests and bug fixes (Briand, *et al*, 1998; Onoma, *et al*, 1995; Singer, 1998; West, 1996). For example, a common denominator in these papers is the emphasis that is placed on a presence of what is termed a bug tracking system, historical data base of changes, or change management. The wording is such that the *internal* use of this information gets emphasized. The information is considered important for the maintainers: they must be able to track similar bugs, they must be able to retrieve the status of each change, and so on. By taking a service perspective, we additionally stress the *external* use, i.e. in the communication with the customer, of essentially the same information in what we call event management.

Stålhane, Borgersen, and Arnesen (1997) did a survey to find those aspects of quality that buyers of software consider most important. The most important result of their study is the strong emphasis customers place on service quality. The top five factors found in their study are: service responsiveness, service capacity, product reliability, service efficiency, and product functionality. They also quote an interesting result from a quality study in the telecommunications domain. On the question ‘Would you recommend others to buy from this company?’, a 100% yes was obtained for the category users that had complained and got a satisfactory result. For the category users that had not complained, this percentage was 87%. Apparently, it is more important to get a satisfactory service than to have no problems at all.

Pitt, Watson, and Kavan also argue that software maintenance has a significant service component. They have used SERVQUAL, an instrument developed in the service marketing area, as a measure of service quality as delivered by IS departments (Pitt, Watson, and Kavan, 1995; Watson, Pitt, and Kavan, 1998).

Literature discussing the practical implementation of a service-oriented maintenance organization is scarce and fragmented. In the next section we discuss two approaches that do offer an overall approach to implementing a service-oriented maintenance organization. One approach is in the form of a set of best practices which is widely used in the Netherlands, the other in the form of a recently developed maturity-growth model. Both these approaches incorporate the service-oriented practices as discussed in section 5. However, as far as we know there is no scientific literature discussing *actual application* of these approaches.

7 IMPLEMENTING A SERVICE-ORIENTED MAINTENANCE ORGANIZATION

If we accept the service perspective on software maintenance, the next logical step is to implement the processes as described in section 5. In this section we discuss two overall approaches for doing so. Both approaches have a scope which is wider than mere software maintenance. An IT service organization may also maintain hardware configurations, handle software distribution, run a computer centre, and so on. Hence, the consequent use of the term service, rather than software maintenance, in the remainder of this section.

The IT Infrastructure Library essentially is a set of best practices. The IT Service Capability Maturity Model is a growth model akin to the Software CMM. At the level of individual processes, these models have a lot in common.

7.1 IT Infrastructure Library

According to CCTA (1993), the primary objective of the IT Infrastructure Library is ‘to establish best practices and a standard of IT service quality that customers should demand and providers should seek to supply.’ ITIL was originally developed by the British government through their Central Computer & Telecommunications Agency (CCTA). Nowadays, ITIL is being maintained by the Netherlands IT Examinations Institute (EXIN).

The library consists of several sets of booklets that contain those ‘best practices’ in IT service delivery. The booklets are divided into nine sets. The first six sets are called the IT service provision and IT infrastructure management sets. The other three are called the Environmental sets. These latter three sets cover the environmental infrastructure for IT, such as the building, cabling and service facilities. We will only look at the IT service provision and IT infrastructure management sets. The six sets cover the following practices (each described in a separate booklet):

- The Service Support set covers configuration management, problem management, change management, help desk, and software control and distribution.
- The Service Delivery set covers service level management, capacity management, contingency planning, availability management, and cost management for IT services.
- The Managers' set deals with managing facilities management and customer liaison.
- The Software Support set describes software life-cycle support and testing an IT service for operational use.
- The Computer Operations set covers computer operations management, unattended operating, third party and single source maintenance, and computer installation and acceptance.
- Finally, the Network set describes the management of local processors and terminals.

Each booklet describes the practices in terms of planning; implementation; audits; benefits, cost and possible problems, and tool support. Attention is given to operational procedures, roles, responsibilities, dependencies, support processes, training, etc.

Although the booklets cover a wide range of issues regarding IT services, there are a number of important issues that need more attention. Examples are:

- The specification of service level agreements. Although ITIL does promote the use of SLAs, it does not provide much help on how to develop them.
- The use of service catalogs. ITIL does promote the use of a service catalog to facilitate the communication with the customers, but again does not say much about the contents or how to develop it.
- ITIL implementation. ITIL itself does not provide much information on the best way to implement the different processes and on how to decide on the best order of implementation.
- The distinction between service producing processes and service support processes. In our opinion, ITIL does not clearly distinguish between those two types. For example, the ITIL help desk is both used for communication with the end-users (needed for incident handling) and for user support (a service).

While over the years different companies have been selling services that complement ITIL, such as education, training, and consulting on ITIL implementation, ITIL still lacks an overall approach to the improvement of service processes. Improvement is not an integral part of the library.

7.2 The IT Service Capability Maturity Model

The IT Service Capability Maturity Model (Niessink and van Vliet, 1998) is a maturity growth model akin to the Software Capability Maturity Model (SEI, 1995). The structure of the model is similar to that of the Software CMM, but its application domain is different. Whereas the Software CMM targets software development processes the IT Service CMM targets the processes that are key to producing high quality IT services. Note that the Software CMM is claimed to be suited for both development and maintenance processes, but difficulties implementing the model in a maintenance-only organization were reported by Drew (1992). IT services are provided by operating, managing, installing, or maintaining the information technology of a customer or supporting the users of that technology. So software maintenance is one (subset) of the possible IT services that can be provided.

The IT Service CMM measures the service process maturity of organizations on a five level ordinal scale. The first—initial—level has no associated key process areas. This is the level where all IT service organizations reside that have not implemented the level two key process areas. Level two is the repeatable level. Organizations that have reached level two will be able to repeat earlier successes in similar circumstances. Thus the emphasis of level two is on getting the IT services right for one customer. On level three, the defined level, the service organization has defined its processes and is using tailored versions of these standard processes to deliver the services. By using common organization-wide standard processes, the process capability to deliver services consistently is improved. At level four, the managed level, organizations gain quantitative insight into their service processes and service quality. By using measurements and an organization-wide measurement database organizations are able to set and achieve quantitative quality goals. Finally, at level five, the optimizing level, the entire organization is focused on continuous process and service improvement. Using the quantitative measurements the organization prevents problems from recurring by changing the processes. The organization is able to introduce new technologies and services into the organization in an orderly manner.

More formally, the five maturity levels are defined as follows:

- **Initial level.** The IT service delivery process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort and heroics.
- **Repeatable level.** Basic service management processes are established to track cost, schedule and performance of the IT service delivery. The necessary discipline is in place to repeat earlier successes on projects with similar services and service levels.
- **Defined level.** The IT service processes are documented, standardized, and integrated into standard service processes. IT services are delivered using approved, tailored versions of the organization's standard service processes.
- **Managed level.** Detailed measurements of the IT service delivery process and service quality are collected. Both the service processes and the delivered services are quantitatively understood and controlled.
- **Optimizing level.** Continuous process improvement is enabled by quantitative feedback from the processes and from piloting innovative ideas and technologies.

Below, we discuss the key process areas for level two of the model to show how they implement the processes identified in section 3. The reader is referred to Niessink and van Vliet (1998) for a description of the key process areas of the higher levels.

The key process areas for level two are concerned with establishing the processes that enable the organization to repeat earlier successful services in similar situations. An organization has to implement two kinds of processes at this level. The first category deals with service management: the planning, specification, tracking and evaluation of services. The second category is concerned with service support: processes that support the activities that actually deliver the services.

The management processes on this level look as follows. First, the service provider and the customer draw up an agreement about the services to be delivered, the quality of the services—specified in terms of service level—and the costs of the services (Service Commitment Management). To ensure that the service levels are realistic, the service provider draws up a service plan that shows the feasibility of the service levels (Service Delivery Planning). During service delivery, the service provider tracks the realized service levels and reports these to the customer on a regular basis to demonstrate that the provider has indeed delivered the services against the promised service levels (Service Tracking and Oversight). After a period of service provision, the customer and the service provider review the service level agreement to see whether it still conforms to the IT needs of the customer (Service Commitment Management). Just like the organization draws up a service level agreement with its customer, the organization should also use service level agreements when it delegates parts of the service delivery to third parties (Subcontract Management).

A level two organization needs to implement three support processes. First, almost all IT services concern the management, operation or maintenance of hardware and software components. Therefore, where necessary for consistent service delivery, these components are put under configuration control. This ensures that at all times the status and history of these components is known (Configuration Management). Second, during the period that the services are delivered, events can occur that need to be resolved by the service provider. These events range from simple requests for service to serious incidents that prevent the customer from using its information technology. All these events need to be identified, tracked, resolved and reported to the customer (Event Management). To service the requests and to resolve incidents, changes to the configuration may be necessary. The change requests are evaluated by the configuration control board with respect to the service level agreement and risk for the integrity of the configuration. Only after a change request has been approved by the change control board, will the configuration be changed (Configuration Management). Finally, to ensure the quality of the services, the service provider deploys quality assurance techniques, such as reviews and audits (Service Quality Assurance).

Not accidentally, the IT Service CMM incorporates the processes identified in section 5. Level 2 of the IT Service CMM precisely covers these processes (under slightly more general labels), plus some processes that are essentially copied from the Software CMM (Subcontract Management, Configuration Management and Service Quality Assurance).

The advantage of the IT Service CMM is that it is a growth model. It provides guidance as to which steps to take next. It is also a stricter model in that it focuses on processes that support and manage service delivery. On the negative side, all the critical notes on the Software CMM hold for the IT Service CMM as well; see Fayad and Laitinen (1997) and Ould (1996). The massive attention of organizations to obtain CMM-like certification holds the danger that focus shifts from developing (and maintaining) software to developing processes. A certified organization, however, does not guarantee the quality of the software developed or maintained under it. A mature maintenance process is not a silver bullet. A framed certificate definitely is not.

Like all other process improvement models, the value of the IT Service CMM lies not in its use a checklist to be followed rigidly. Rather, it provides guidance to an organization wanting to improve its IT service processes. As such, experiences with using the IT Service CMM in discussions within maintenance and IT Service organizations have been very successful.

8 CONCLUSIONS

In this paper we have described the differences between products and services in general, and between software maintenance and software development in particular.

We have shown how the differences between services and products affect the way in which customers judge the quality of products and services and hence how these differences affect organizations that maintain software. From these differences we have deduced four processes that pertain to the quality of software maintenance:

- Translate customer expectations with respect to maintenance into clear service agreements.
- Use these service agreements as a basis for planning and implementing maintenance activities.
- Ensure that maintenance is done according to planning and procedures.
- Manage the communication about the maintenance activities carried out.

Software maintenance organizations can improve the quality of their services, not only by improving the technical quality of their work—less faults, better functionality—but also by improving the functional quality of software maintenance—i.e. by implementing the above processes.

The literature on software maintenance as a service is scarce and fragmented. We know of two overall approaches to implementing a service-oriented maintenance organization: ITIL and the IT Service CMM. However, we know of no scientific literature on the application of ITIL and the IT Service CMM is still in its infancy. We have had some encouraging experiences with using the IT Service CMM in two process assessments aimed at improving software maintenance organizations. These assessments have shown that the service-related key process areas of the IT Service CMM address essential success factors of mature service organizations. Our current research is concerned with the further application and validation of this model.

Acknowledgments

This research was partly supported by the Dutch Ministry of Economic Affairs, project ‘KWINTES’, nr. ITU-96024. Partners in this project are Cap Gemini, Twijnstra Gudde, the Tax and Customs Computer and Software Centre of the Dutch Tax and Customs Administration, and the Technical Universities of Delft and Eindhoven.

The case studies described in section 4 were done by Jeroen Bögels, Lennart Schaftenaar, and Mark van der Zwan.

We acknowledge the useful comments given by Shari Lawrence Pfleeger on a previous version of this paper.

CMM and Capability Maturity Model are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

References

Berry, Leonard L.; and Parasuraman, A. (1991) *Marketing Services—Competing Through Quality*, The Free Press, Macmillan Inc., New York NY, 212 pp.

Briand, Lionel; Kim, Yong-Mi; Melo, Walcèlio; Seaman, Carolyn; and Basili, Victor R. (1998) ‘Q-MOPP: qualitative evaluation of maintenance organizations, processes and products’, *Journal of Software Maintenance*, **10**(4), pp. 249-278.

CCTA (1993) *The IT Infrastructure Library—An Introduction*, Central Computer and Telecommunications Agency (CCTA), HMSO Books, Norwich, England, 45 pp.

Drew, Daniel W. (1992) ‘Tailoring the Software Engineering Institute’s (SEI) Capability Maturity Model (CMM) to a Software Sustaining Engineering

Organization', in *Proceedings of the Conference on Software Maintenance,* IEEE Computer Society Press, Los Alamitos, CA, pp. 137-144.

Fayad, Mohamed E.; and Laitinen, Mauri (1997) 'Process assessment considered wasteful', *Communications of the ACM,* **40**(11), pp. 125-128.

Grönroos, Christian (1990) *Service Management and Marketing—Managing the Moments of Truth in Service Competition*, Lexington Books, Lexington, Mass., 298 pp.

IEEE (1990) *IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12),* Institute of Electrical and Electronics Engineers, Inc., New York, NY., 83 pp.

Lientz, B.P.; and Swanson, E.B. (1980*) Software Maintenance Management: A Study of the Maintenance of Computer Application Software in 487 Data Processing Organizations,* Addison-Wesley Publishing Company, Reading, Mass, pp. ??.

Niessink, Frank; and Vliet, Hans van (1998) 'Towards Mature IT Services', *Software Process—Improvement and Practice*, **4**(2), pp. 55-71.

Onoma, A.K.; Tsai, W.T.; Tsunoda, F.; Sugunuma, H.; and Subramanian, S. (1995) 'Software Maintenance—an Industrial Experience', *Journal of Software Maintenance,* **7**(?) pp. 333-375.

Ould, Martyn A. (1996) 'CMM and ISO 9001', *Software Process—Improvement and Practice,* **2**(4), pp. 281-289.

Parasuraman, A.; Zeithaml, V.A.; and Berry, L.L. (1985) 'A Conceptual Model of Service Quality and its Implication for Future Research', *Journal of Marketing,* **49**(Fall), pp. 41-50.

Pigoski, Thomas M. (1996) *Practical Software Maintenance: Best Practices for Managing Your Software Investment*, John Wiley&Sons, New York, NY., 384 pp.

Pitt, Leyland F.; Watson, Richard T.; and Kavan, C. Bruce (1995) 'Service Quality: A Measure of Information Systems Effectiveness', *Management Information Systems Quarterly,* **19**(2), pp. 173-187.

SEI (1995) *The Capability Maturity Model: Guidelines for Improving the Software Process,* Software Engineering Institute, Carnegie Mellon University, SEI Series in Software Engineering, Addison-Wesley Publishing Company, Reading, Mass., 441 pp.

Singer, Janice (1998) 'Practices of Software Maintenance', in *Proceedings of the International Conference on Software Maintenance*, IEEE Computer Society Press, Los Alamitos, CA., pp. 139-145.

Stålhane, T.; Borgersen, P.C.; and Arnesen, K. (1997) 'In Search of the Customer's Quality View', *The Journal of Systems and Software*, **38**(1), pp. 85-93.

Stark, George E.; and Oman, Paul W. (1997) 'Software maintenance management strategies: observations from the field', *Journal of Software Maintenance*, **9**(6), pp. 365-378.

Trienekens, Jos; Zwan, Mark van der; and Bouman, Jacques (1999) 'Specification of Service Level Agreements, clarifying concepts on the basis of practical research', in *Proceedings of the Software Technology and Engineering Practice Conference,* IEEE Computer Society Press, Los Alamitos, CA., pp. 169-178.

Watson, Richard T.; Pitt, Leyland F.; and Kavan, C. Bruce (1998) ‘Measuring Information Systems Service Quality: Lessons from Two Longitudinal Case Studies’, *Management Information Systems Quarterly*, **22**(1), pp. 61-79.

West, Richard (1996) ‘Book Review: Improving the Maintainability of Software’, *Journal of Software Maintenance*, **8**(5), pp. 345-356.

Zeithaml, Valarie A.; and Bitner, Mary Jo (1996) *Service Marketing*, McGraw-Hill, New York, NY, 700 pp.

Author biographies

Drs. **Frank Niessink** is a PhD candidate at the Vrije Universiteit. His research interests are software measurement, software maintenance and process improvement. During his PhD research, he was involved in two national research projects on IT services. Frank Niessink received a MSc in Computer Science and a MSc in Economics from the Vrije Universiteit.

Prof.dr. **Hans van Vliet** is a Professor of Computer Science at the Vrije Universiteit. His research interests include software measurement and software architecture. Before joining the VU, he worked as a researcher at the Centrum voor Wiskunde en Informatica (Amsterdam) and he spent a year as a visiting researcher at the IBM Research Laboratory in San Jose, CA. Hans van Vliet has a MSc in Computer Science from the Vrije Universiteit and a Ph.D. degree in Computer Science from the University of Amsterdam.

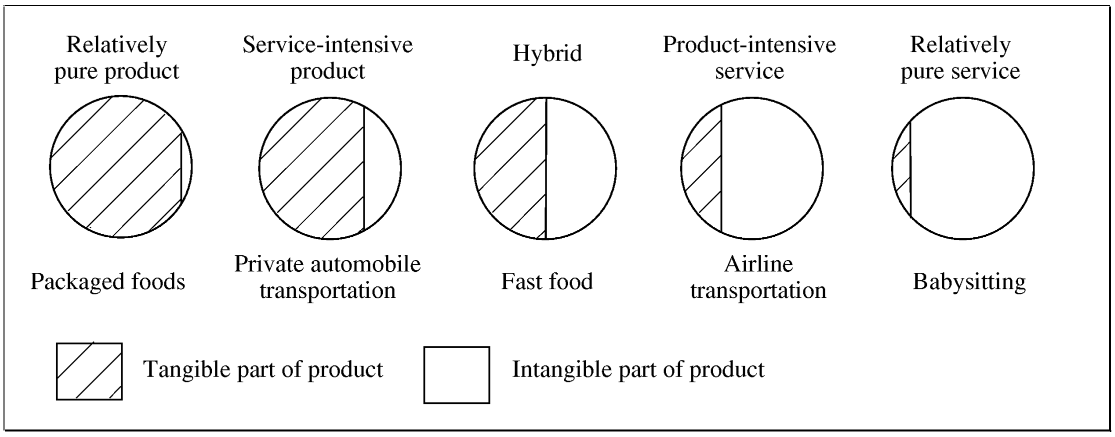

Figure 1. The product-service continuum (Berry and Parasuraman, 1991, p. 9)

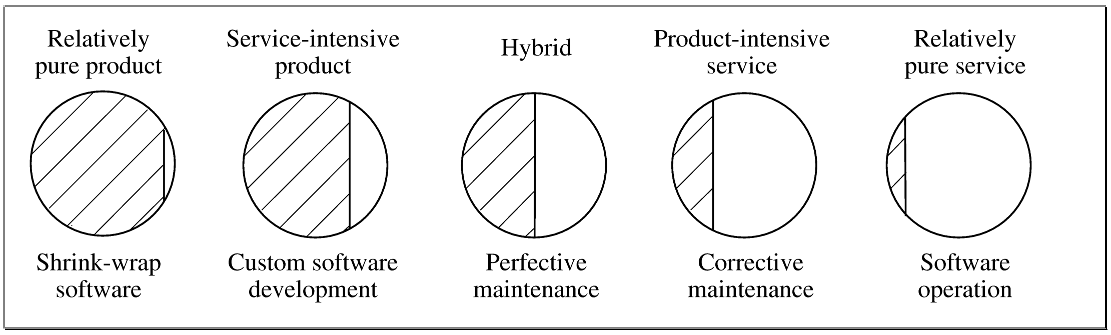

Figure 2. The product-service continuum for software development and maintenance

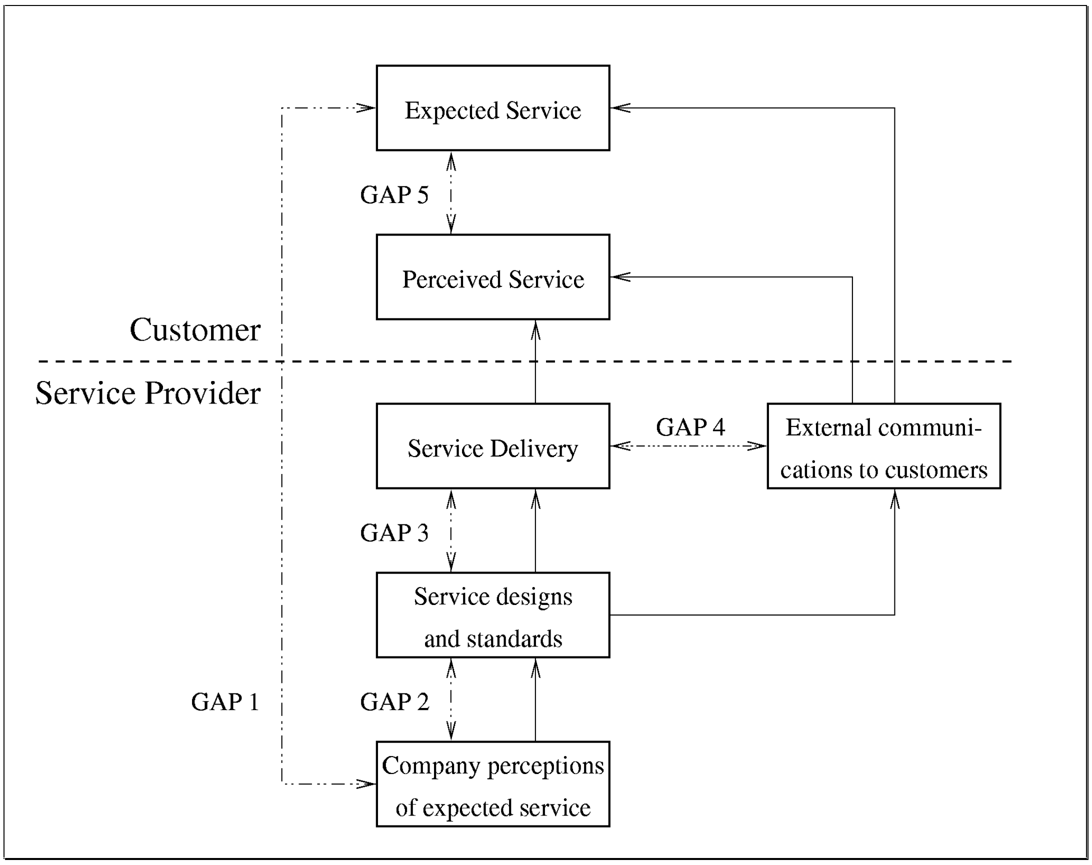
Expected Service
GAP 5
Perceived Service
Customer
Service Provider
Service Delivery
GAP 4
External communi-cations to customers
GAP 3
Service designs and standards
GAP 1
GAP 2
Company perceptions of expected service

Figure 3. The Gaps model of service quality (Parasuraman, Zeithaml and Berry, 1985)